KAKATIYA UNIVERSITY WARANGAL

Under Graduate Courses (Under CBCS with effect from Academic Year 2022-2023 onwards)

B.Sc. DATA SCIENCE

III Year: Semester-V

Paper – V (A): Natural Language Processing

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to NLP. It deals with morphological processing, syntactic parsing, information extraction, probabilistic NLP and classification of text using Python's NLTK Library.

Outcomes:

At the end of the course the student will be able to

- Write Python programs to manipulate and analyze language data
- Understand key concepts from NLP and linguistics to describe and analyze language
- Understand the data structures and algorithms that are used in NLP
- Classify texts using machine learning and deep learning

Unit-I

Language Processing and Python: Computing with Language: Texts and Words, A Closer Look at Python: Texts as Lists of Words, Computing with Language: Simple Statistics, Back to Python: Making Decisions and Taking Control, Automatic Natural Language Understanding [Reference 1]
Accessing Text Corpora and Lexical Resources: Accessing Text Corpora, Conditional Frequency Distributions, Lexical Resources, WordNet [Reference 1]

Unit-II

Processing Raw Text: Accessing Text from the Web and from Disk, Strings: Text Processing at the Lowest Level, Text Processing with Unicode, Regular Expressions for Detecting Word Patterns, Useful Applications of Regular Expressions, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation, Formatting: From Lists to Strings. [Reference 1]
Categorizing and Tagging Words: Using a Tagger, Tagged Corpora, Mapping Words to Properties Using Python Dictionaries, Automatic Tagging, N-Gram Tagging, Transformation-Based Tagging, How to Determine the Category of a Word [Reference 1]

Unit-III

Learning to Classify Text: Supervised Classification, Evaluation, Naive Bayes Classifiers [Reference 1]
Deep Learning for NLP: Introduction to Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks, Classifying Text with Deep Learning [Reference 2]

Unit-IV

Extracting Information from Text

Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction. [Reference 1]

Analyzing Sentence Structure

Some Grammatical Dilemmas, What's the Use of Syntax. Context-Free Grammar, Parsing with Context-Free Grammar, [Reference 1]

References:

1. Natural Language Processingwith Python. Steven Bird, Ewan Klein, and Edward Lope, O'Reily, 2009
2. Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. AkshayKulkarni,AdarshaShivananda, Apress, 2019

Suggested Reading:

3. Allen James, Natural Language Understanding, Benjamin/Cumming,1995.
4. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.

Practical – 5(A): Natural Language Processing (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objective: The main objective of this laboratory is to write programs that manipulate and analyze language data using Python

This lab requires mentoring sessions from TCS.

Python Packages

Students are expected to know/ learn the following PythonNLP packages

- NLTK (www.nltk.org/ (http://www.nltk.org/))
- Spacy (https://spacy.io/)
- TextBlob (http://textblob.readthedocs.io/en/dev/
- Gensim (https://pypi.python.org/pypi/gensim)
- Pattern (https://pypi.python.org/pypi/Pattern)

Datasets:

1. NLTK includes a small selection of texts from the Project Gutenberg electronic text archive, which contains some 25,000 free electronic books, hosted at *http://www.gutenberg.org/*.
2. The Brown Corpus contains text from 500 sources, and the sources have been categorized by genre, such as *news*, *editorial*, and so on (*http://icame.uib.no/brown/bcm-los.html*).
3. Wikipedia Articles Or any other dataset of your choice

Reference:

Jacob Perkins. Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing. 2014

Exercises:

1. Text segmentation: Segment a text into linguistically meaningful units, such as paragraphs, sentences, or words. Write programs to segment text (in different formats) into tokens (words and word-like units) using regular expressions. Compare an automatic tokenization with a gold standard
2. Part-of-speech tagging: Label words (tokens) with parts of speech such as noun, adjective, and verb using a variety of tagging methods, e.g., default tagger, regular expression tagger, unigram tagger, and n-gram taggers.
3. Text classification: Categorize text documents into predefined classes using Naïve Bayes Classifier and the Perceptron model
4. Chunk extraction, or partial parsing: Extract short phrases from a part-of-speech tagged sentence. This is different from full parsing in that we're interested in standalone chunks, or phrases, instead of full parse trees

5. Parsing: parsing specific kinds of data, focusing primarily on dates, times, and HTML. Make use of the following preprocessing libraries:
   - dateutil which provides datetime parsing and timezone conversion
   - lxml and BeautifulSoup which can parse, clean, and convert HTML
   - charade and UnicodeDammit which can detect and convert text character encoding
6. Sentiment Analysis: Using Libraries TextBlob and nltk, give the sentiment of a document